1. (Trappe, page 159: 1) The ciphertext 5859 was obtained from the RSA algorithm using \( n = 11413 \) and \( e = 7467 \). Using the factorization \( 11413 = 101 \cdot 113 \), find the plaintext

**Sol:**

\[
\phi(n) = (p-1) \cdot (q-1) = 100 \cdot 112 = 11200
\]

\[
d \equiv e^{-1} \equiv 3 \pmod{11200} \text{ (using extended Euclidean algorithm)}
\]

\[
m \equiv e^d \equiv 5859^3 \equiv 1415 \pmod{11413}
\]

verification: \( c \equiv 1415^{7467} \equiv 5859 \pmod{11413} \)

2. (Trappe, page 159: 4) Naïve Nelson uses RSA to receive a single ciphertext \( c \), corresponding to the message \( m \). His public modulus is \( n \) and his public encryption exponent is \( e \). Since he feels guilty that his system was used only once, he agrees to decrypt any ciphertext that someone sends him, as long as it is not \( c \), and return the answer to that person. Evil Eve sends him the ciphertext \( 2^e c \pmod{n} \). Show how this allows Eve to find \( m \).

**Sol:**

Nelson checks that \( 2^e c \) is not \( c \), then decrypts it and gets

\[
(2^e c)^d \equiv 2c \equiv 2m \pmod{n}
\]

Eve divides this by 2 and obtains the original message \( m \).

3. (Trappe, page 159: 5) In order to increase security, Bob chooses \( n \) and two encryption exponents \( e_1, e_2 \). He asks Alice to encrypt her message \( m \) to him by first computing \( c_1 \equiv m^{e_1} \pmod{n} \), then encrypting \( c_1 \) to get \( c_2 \equiv c_1^{e_2} \pmod{n} \). Alice then sends \( c_2 \) to Bob. Does this double encryption increase security over single encryption? Why or why not?

**Sol:**

Alice encrypts her message \( m \) as \( c_2 \equiv c_1^{e_2} \equiv m^{e_1 e_2} \pmod{n} \)

To decrypt this message, Bob calculates \( (c_2^{d_2})^{d_1} \pmod{n} \), where \( e_1 \cdot d_1 \equiv 1 \pmod{\phi(n)} \) and \( e_2 \cdot d_2 \equiv 1 \pmod{\phi(n)} \).

However, the above procedure is equivalent to the following

Alice encrypts her message \( m \) as \( c_2 \equiv m^{e_3} \pmod{n} \) and

Bob decrypts the ciphertext as \( m \equiv c_2^{d_3} \pmod{n} \) where

\( e_1 \cdot e_2 \equiv e_3 \pmod{\phi(n)} \) and \( d_1 \cdot d_2 \equiv d_3 \pmod{\phi(n)} \)

(which naturally leads to \( e_3 \cdot d_3 \equiv 1 \pmod{\phi(n)} \) and \( c_2^{d_3} \equiv (m^{e_3})^{d_3} \equiv m \pmod{n} \))
Therefore, this scheme is equivalent to a simple RSA scheme with a different key pairs \((n, e_3), (n, d_3)\). The security is not enhanced.

4. (Trappe, page 159: 6) Let \(p\) and \(q\) be distinct odd primes, and let \(n = p \cdot q\).
Suppose that the integer \(x\) satisfies \(\gcd(x, p \cdot q) = 1\).

(a) Show that \(x^{\phi(n)/2} \equiv 1 \pmod{p}\) and \(x^{\phi(n)/2} \equiv 1 \pmod{q}\).

(b) Use (a) to show that \(x^{\phi(n)/2} \equiv 1 \pmod{n}\).

(c) Use (b) to show that if \(e \cdot d \equiv 1 \pmod{\phi(n)/2}\) then \(x^{e \cdot d} \equiv x \pmod{n}\) (This shows that we could work with \(\phi(n)/2\) instead of \(\phi(n)\) in RSA.)

**Sol:**

(a)

\[
\phi(n)/2 = (p-1) \cdot (q-1)/2 \\
2 \mid (q-1) \Rightarrow (p-1) \mid \phi(n)/2 \Rightarrow \phi(n)/2 = k \cdot (p-1) \\
\text{therefore, } x^{\phi(n)/2} = (x^k)^{p-1} \equiv 1 \pmod{p} \text{ from Fermat theorem} \\
\text{this is also true for } x^{\phi(n)/2} \equiv 1 \pmod{q}
\]

(b) using CRT,

\[x^{\phi(n)/2} \equiv 1 \pmod{p} \text{ and } x^{\phi(n)/2} \equiv 1 \pmod{q} \text{ implies } x^{\phi(n)/2} \equiv 1 \pmod{n} \]

Actually, this can also be derived from Carmichael theorem

\[x^{\lambda(n)} \equiv 1 \pmod{n} \text{ where } \lambda(n) = \text{lcm}(p-1, q-1)\]

since

\[\phi(n) = (p-1) \cdot (q-1) = \gcd(p-1, q-1) \cdot \text{lcm}(p-1, q-1) \text{ and} \]

\[2 \mid \gcd(p-1, q-1) \Rightarrow \text{lcm}(p-1, q-1) \mid \phi(n)/2 \]

i.e.

\[\text{lcm}(p-1, q-1) \mid \phi(n)/2\]

(c) if \(e \cdot d \equiv 1 \pmod{\phi(n)/2}\) then

\[e \cdot d = 1 + k \cdot \phi(n)/2 \Rightarrow x^{e \cdot d} = x^{1 + k \cdot \phi(n)/2} \equiv x \pmod{n}\]

from the result of (b)

5. (Trappe, page 160: 12) Show that if \(x^2 \equiv y^2 \pmod{n}\) and \(x T \pm y \pmod{n}\), then \(\gcd(x+y, n)\) is a nontrivial factor of \(n\).

**Sol:**

We prove this in a way exactly the same as we prove the basic principle of factoring.
let \( d = \gcd(x+y, n) \)

There are three cases to be discussed:
Case 1 \( d = n \): if \( d = n \), then \( x+y \equiv 0 \pmod{n} \), contradiction.
Case 2 \( d = 1 \):
\[
x^2 \equiv y^2 \pmod{n} \Rightarrow x^2 - y^2 = (x-y)(x+y) = k \cdot n
\]
\( \Rightarrow x-y \equiv 0 \pmod{n} \), contradiction

6. (Trappe, page 162: 4) Factor 618240007109027021 by the p-1 method

**Sol:**

Because this number is bigger than \( 2^{32} \), MATLAB cannot process this number directly. We are going to use MAPLE kernel in MATLAB for solving this factoring problem with p-1 factoring method.

In a p-1 factoring method, we hope that there is discrepancy in the factors of p-1 and q-1.

If we can find a number, say \( B! \), such that \( p-1 \mid B! \) but \( q-1 \nmid B! \), then \( a^{B!} \equiv 1 \pmod{p} \) but \( a^{B!} \equiv 1 \pmod{q} \), i.e. \( a^{B!} - 1 = k \cdot p \neq k' \cdot q \). Therefore, \( a^{B!-1} \pmod{n} \) is not 0 and \( \gcd(a^{B!-1} \pmod{n}, n) = p \) is a nontrivial factor of n.

```maple
>> maple('n:=618240007109027021; B:=5; p:=gcd((5&^factorial(B)) mod n -1, n)')
an=
n := 618240007109027021  B := 5  p := 1
>> maple('n:=618240007109027021; B:=10; p:=gcd((5&^factorial(B)) mod n -1, n)')
an=
n := 618240007109027021  B := 10  p := 1
>> maple('n:=618240007109027021; B:=15; p:=gcd((5&^factorial(B)) mod n -1, n)')
an=
n := 618240007109027021  B := 15  p := 1
>> maple('n:=618240007109027021; B:=20; p:=gcd((5&^factorial(B)) mod n -1, n)')
an=
n := 618240007109027021  B := 20  p := 1
>> maple('n:=618240007109027021; B:=25; p:=gcd((5&^factorial(B)) mod n -1, n)')
an=
n := 618240007109027021  B := 25  p := 250387201
>> maple('n:=618240007109027021; B:=30; p:=gcd((5&^factorial(B)) mod n -1, n)')
an=
n := 618240007109027021  B := 30  p := 250387201
```

As a comparison, we factors 618240007109027021 directly with ifactor():

```maple
> maple('ifactor(618240007109027021)')
an = ``(2469135821)*``(250387201)
```

If you have the MAPLE software installed, you can write a for loop to perform the above search procedure, the codes looks like:

```
n := 618240007109027021;
B := 1;
```
7. (Trappe, page 162: 6) Let $n = 537069139875071$. Suppose you know that

$$85975324443166^2 \equiv 462436106261^2 \pmod{n}$$

Factor $n$.

**Sol:**

Clearly $85975324443166 \pm 462436106261 \pmod{n}$ because

$$-462436106261 \equiv 536606703768810 \pmod{n}$$

$\text{gcd}(85975324443166-462436106261, 537069139875071) = 9876469$

$\text{gcd}(85975324443166+462436106261, 537069139875071) = 54378659$

while $9876469 \times 54378659 = 537069139875071$

the above can be easily done through MAPLE

```maple
>> maple('n:= 537069139875071; b:=n - 462436106261:')
ans =  n := 537069139875071  b := 536606703768810
>> maple('c=gcd(537069139875071, 85975324443166-462436106261)')
ans =  c = 9876469
>> maple('c=gcd(537069139875071, 85975324443166+462436106261)')
ans =  c = 54378659
>> maple('c=9876469*54378659')
ans =  c = 537069139875071
```